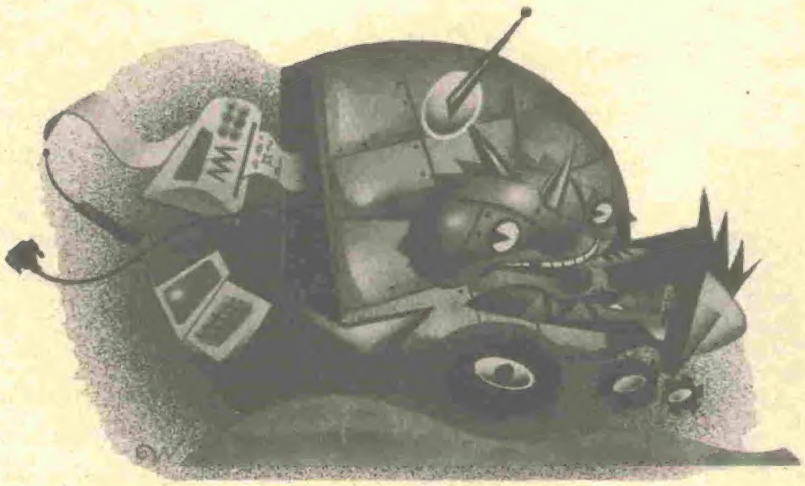
VISUAL BASIC

ศรันย์ ไมตรีเวช

พื้นฐานการติดต่อกับดาต้าเบส ด้วยวิชวลเบสิก 4.0



การติดต่อกับดาต้าเบสด้วย วิชวลเบสิก 4.0 นั้นง่ายกว่า เวอร์ชันก่อนมาก แต่ก็มีอะไร หลายอย่างที่คุณต้องเรียนรู้เพิ่มเติม มาดูกันว่าเราควรดูอะไรก่อน

สิ่งหนึ่งที่ไมโครซอฟต์ได้จากการเปิดกว้างให้บริษัทซอฟต์แวร์อื่นเข้าร่วมประสานงานด้วย เช่น การเปิดโอกาสให้ใครต่อใครสร้างคัสตอมคอนโทรลเพื่อนำมาใช้ร่วมกับวิชวลเบสิกนั้น ทำให้ไมโครซอฟต์เกิดไอเดียหลากหลายขึ้น เข้าใจความต้องการของผู้ใช้มากขึ้น และที่สุดแล้วก็คือ นำมาซึ่งการปรับปรุงพัฒนาเวอร์ชันใหม่ ๆ ได้ตรงตามใจตลาดกว่าเดิม

เวอร์ชัน 4.0 ของวิชวลเบสิกนี้ไมโครซอฟต์ไม่สามารถก๊กฟีเจอร์ดี ๆ ไว้ โดยปล่อยให้เป็นหน้าที่ของบริษัทซอฟต์แวร์อื่น หรือไม่ก็ทำออกมาจำหน่ายเองในภายหลัง เคลไฟและพาวเวอร์บิลเดอร์ให้อะไรต่ออะไรกับผู้ใช้พร้อมสรรพ ล้วนแล้วแต่ของดี ๆ น่าใช้ทั้งนั้น หากจะประคองความรู้สึกของผู้ใช้ให้ติดอยู่กับวิชวลเบสิก ก็จำเป็นต้องมีของดีให้มากกว่าที่เคยพอสมควร

ตัวอย่างที่เห็นได้ชัดคือเครื่องมือในการติดต่อกับดาต้าเบสชิ้นสำคัญ เช่น Grid นั้น เมื่อเวอร์ชันก่อนไมโครซอฟต์ทำเป็นไม่รู้ไม่ชี้ ปล่อยให้โปรแกรมเมอร์ลำบากลำบนกับการเขียนโค้ดเอง เพื่อนำข้อมูลจากดาต้าคอนโทรลหรือออปเจ็กต์ดาต้าเบสขึ้นมาแสดงใน Grid เนื่องจาก Grid นั้นสายตาสั้น มองไม่เห็น Data Control มีแต่คอนโทรลพื้น ๆ อย่าง TextBox, Label หรือ Image ที่มองเห็น ทั้งที่จริงแล้วในการพัฒนาแอพพลิเคชันเพื่อติดต่อกับดาต้าเบสทั่วไปนั้น การนำเสนอข้อมูลในรูปแบบของตาราง Grid เป็นสิ่งปฏิบัติกันเป็นสามัญ เรียกได้ว่าจำเป็นอย่างยิ่งยวด

ไมโครซอฟต์ปล่อยให้ซอฟต์แวร์เช่น Q+E, Spread, TrueGrid หรือ Formula I รับผิดชอบเรื่องนี้ไป ใครอยากนำเสนอข้อมูลปริมาณมากในรูปของตารางด้วยวิธีง่าย ๆ ก็ต้องไปซื้อหาคัสตอมคอนโทรลเหล่านี้มาใช้ หาไม่แล้วก็จงก้มหน้าก้มตาเขียนโปรแกรมเรียกข้อมูลมาใส่ตาราง Grid ต่อไป

เวอร์ชัน 4.0 นี้ ไมโครซอฟต์ให้คัสตอมคอนโทรล ซึ่งจะถูกโหลดขึ้นมาโดยดีฟอลต์อยู่สามตัวที่เกี่ยวกับการแสดงข้อมูลจำนวนมากพร้อมกันทีเดียว นั่นคือ DBGrid, DBList และ DBCombo (สำหรับ DBGrid ซื้อมาจาก Apex นะครับ ส่วน DBList กับ DBCombo เป็นของไมโครซอฟต์เอง เรียกว่าขอยืนครึ่งเดียวพอ) ความจริงทั้ง Grid, ListBox และ ComboBox ก็ยังมีให้คุณใช้เหมือนเดิม ไม่ได้หายหนไปไหน กลุ่มคอนโทรลที่นำชื่อด้วยคำว่า 'DB' เพียงมีคุณสมบัติพิเศษเหนือกว่าเข้าถึงดาต้าเบสได้สะดวกกว่า และแน่นอน กินทรัพยากรเยอะกว่า อันนี้เป็นทางเลือกว่าหากต้องการแสดงข้อมูลที่ไม่เกี่ยวกับดาต้าเบส ก็ไม่ต้องใช้คอนโทรลกลุ่ม DB ให้เปลืองทรัพยากรเปล่า

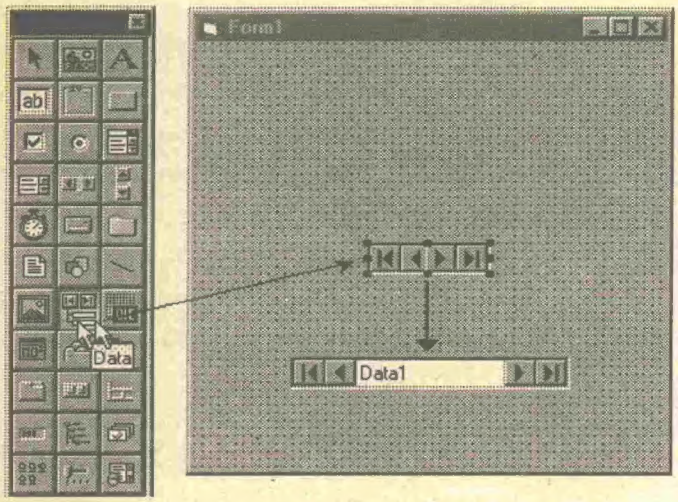
สำหรับตอนนี้ผมจะเสนอคุณสมบัติและการใช้ที่น่าสนใจของคอนโทรลกลุ่ม DB รวมทั้งตัว Data Control

เอง ซึ่งมีความแตกต่างจากของเดิมพอควร มาเริ่มจาก Data Control ไล่ไปหา DBGrid, DBList และ DBCombo ตามลำดับเลยครับ

Data Control

วิธีสร้างขึ้นบนฟอร์ม

ดับเบิลคลิ๊กไอคอนบนทูลบ็อกซ์ดังรูปที่ 1 จะเห็น Data Control ปรากฏขึ้นบนฟอร์ม ควรใช้เม้าส์ขยายออกตามแนวนอนให้ได้ขนาดที่เหมาะสมหากคิดจะให้ผู้ใช้ดูข้อมูลได้ทั่วถึงด้วยบาร์เลื่อนเรคอร์ด



รูปที่ 1 การสร้าง Data Control ขึ้นบนฟอร์ม

หน้าที่หลัก

เข้าถึงดาต้าเบสที่คุณระบุ โดยจะเข้าถึงได้เพียงครั้งละหนึ่งกลุ่มเรคอร์ด กลุ่มเรคอร์ดดังกล่าวนี้อาจเป็นเทเบิลใดเทเบิลหนึ่งทั้งตัว หรือไม่ก็เป็นกลุ่มเรคอร์ดจากหลายเทเบิลผสมกันตามการตั้งเงื่อนไขเรียกข้อมูล (query) ของคุณ

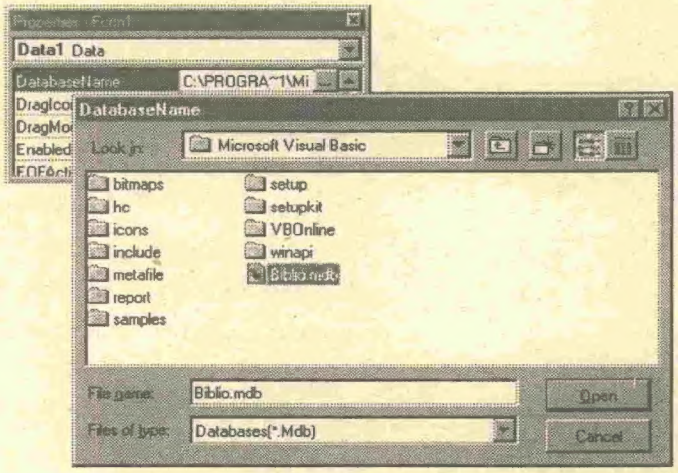
กลุ่มเรคอร์ดดังกล่าวได้มาโดยผ่านออปเจ็กต์ Recordset ออปเจ็กต์ชนิดนี้มีโครงสร้างเป็นลักษณะหลายคอลัมน์หลายบรรทัด ซึ่งนั่นก็หมายความว่า มันสามารถบรรจุข้อมูลหลายเรคอร์ดและหลายฟิลด์ได้ทีเดียวพร้อมกัน Recordset มีอยู่ 3 ชนิด คือ Table-type, Dynaset-type และ Snapshot-type สำหรับวิชวลเบสิกเวอร์ชัน 3.0 นั้น Data Control ใช้ออปเจ็กต์ Recordset ในการเข้าถึงข้อมูลได้เพียงลักษณะชนิด Dynaset เท่านั้น แต่เวอร์ชัน 4.0 นี้จะเป็นไปได้ทั้งสามลักษณะ

การแสดงข้อมูลนั้นทำไม่ได้ที่ตัว Data Control เอง เนื่องจาก Data Control ไม่ได้มีส่วนของการแสดง เช่น รูปตารางหรือรูปช่องว่างใด ๆ ดังนั้นในทางปฏิบัติเราจึงใช้ Data Control เพียงเพื่อเข้าถึงข้อมูล ส่วนการรับและแสดงข้อมูลให้ปรากฏต่อตาผู้ใช้นั้น ต้องกระทำผ่านสิ่งที่เรียกว่าเบาวน์ดคอนโทรล (Bound Controls) ซึ่งก็คือคอนโทรลที่สามารถรับรู้ข้อมูล (Data-Aware Controls) ที่ถ่ายทอดหรือส่งผ่านข้อมูลให้ Data Control ได้ คุณจะรู้ว่าคอนโทรลตัวไหนสามารถเป็นเบาวน์ดคอนโทรลได้ก็จากการดูง่าย ๆ จากพรอพเพอร์ตี้ส์ DataSource หากคอนโทรลใดมีพรอพเพอร์ตี้ส์นี้ คอนโทรลนั้นแหละครับมีศักยภาพที่จะเป็นเบาวน์ดคอนโทรลได้

พรอพเพอร์ตี้ส์หลัก

Connect บ่งบอกชนิดของดาต้าเบส โดยดีฟอลต์จะเป็น Access หากเป็นเบสอื่นก็เลือกได้จากดร็อบดาวน์ลิสต์ ซึ่งอยู่ในช่อง Value เช่น หากเป็นเบส FoxPro ค่าของ Connect ก็จะเป็น FoxPro 2.0, FoxPro 2.5 หรือ FoxPro 2.6 มีข้อที่ต้องหมายเหตุไว้คือ ถ้าเป็นเบสของ Paradox คุณต้องใส่รหัสผ่านพ่วงท้ายเข้าไปด้วย เช่น Data1.Connect=paradox 3.x;pwd=MyPassword

DatabaseName กำหนดว่าจะให้ Data Control อ่านข้อมูลจากดาต้าเบสไหน หากเป็นเบสของแอ็กเซส คุณต้องหาชื่อไฟล์ที่มีนามสกุล .MDB มาใส่ วิธีง่ายที่สุดคือคลิ๊กปุ่มสามจุดด้านขวามือของช่องรับ Value คุณจะเห็นหน้าต่างเลือกดาต้าเบสปรากฏขึ้นมาดังรูปที่ 2 (ผมจะใช้ดาต้าเบส Biblio.MDB ซึ่งมีอยู่แล้วในไดเรคทอรีของวิชวลเบสิกสาธิตนับจากนี้ หากคุณกำลังเปิดวิชวลเบสิกพร้อมอ่านไปด้วยก็เลือกดาต้าเบสนี้ตามก็แล้วกัน) หากต้องการเปลี่ยนดาต้าเบสขณะรัน หรือพูดง่าย ๆ ว่าต้องเขียนโค้ดเพื่อกำหนดชื่อดาต้าเบสเสียใหม่ สิ่งที่คุณต้องไม่ลืม (แต่มักจะลืม) คือพาธที่อยู่ของไฟล์ด้วย เช่น Data1.



รูปที่ 2 หน้าต่างเลือกดาต้าเบส

DatabaseName="C:\Data\MyBase.MDB" แต่ถ้าหากเบสเป็นชนิดอื่น ก็ต้องระบุเป็นชื่อพาธ เช่น Data1.DatabaseName="C:\Data\" เท่านี้ โดยคุณต้องแน่ใจว่าไดเรคทอรี Data นั้นจะต้องมีกลุ่มเทเบิลอันเป็นตัวตนของดาต้าเบสที่ต้องการ

มีหมายเหตุไว้สำหรับเบสชนิด Btrieve นั่นคือคุณจะต้องระบุชื่อไฟล์ซึ่งมีนามสกุล .DDF ไว้ด้วย เช่น Data1.DatabaseName="C:\Data\MyBase.DDF"

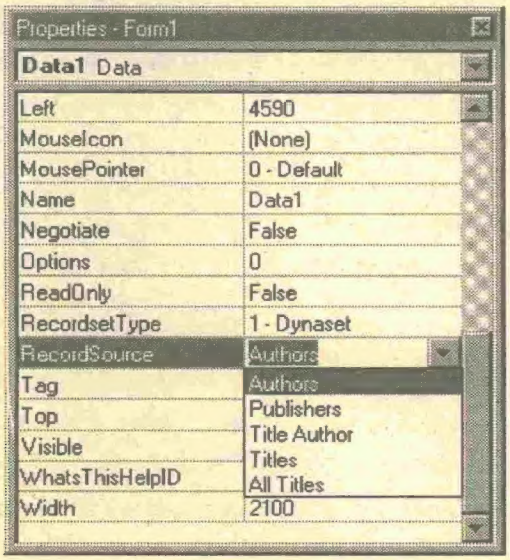
RecordSource กำหนดเทเบิลหรือเงื่อนไขการเรียกข้อมูล (รวมทั้ง QueryDef ซึ่งเก็บในดาต้าเบสเหมือนเทเบิล แต่ความจริงเป็นคำสั่งเรียกข้อมูลแบบกำหนดเงื่อนไข) หากคุณเลือก DatabaseName ไว้แล้ว ก็มีสิทธิ์เลือก RecordSource ได้ทันทีในหน้าต่างพรอพเพอร์ตี้ส์ขณะอยู่ในโหมดก่อนรัน อย่างที่เห็นในรูปที่ 3 ที่เห็นนั้นเป็นเทเบิลภายในดาต้าเบส Biblio.MDB ซึ่งจะลิสต์ลงมาให้เลือกทันทีที่คุณคลิ๊กปุ่มชี้ลงด้านขวามือของช่องรับ Value และหากต้องการเขียนโค้ดเพื่อเปลี่ยนเทเบิลก็สามารถทำได้ เช่น Data1. RecordSource= "Titles"

```
Data1.Refresh
```

หรือ

```
Data1.RecordSource="Select * from Titles Where
[Year Published] >=10000"
Data1.Refresh
```

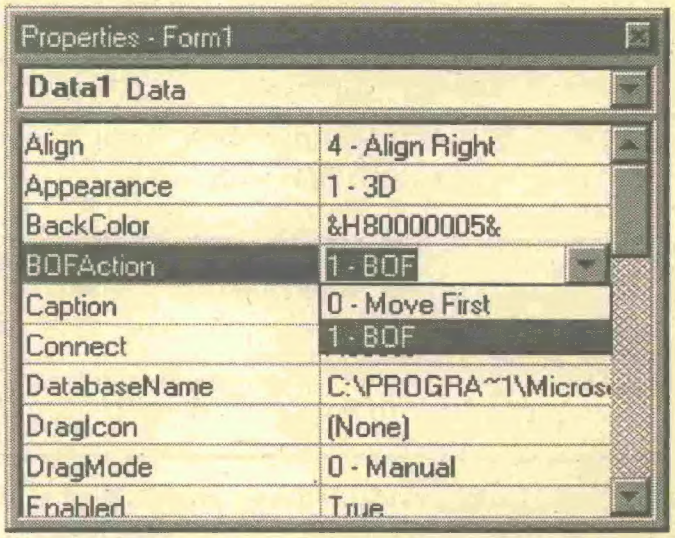
และต้องหมายเหตุไว้ด้วยว่า หากเขียนโค้ดเพื่อเปลี่ยนเทเบิลขณะรัน คุณต้องใช้วิธีการ Refresh ของ Data Control ในบรรทัดต่อมาเพื่อให้เกิดผลตามต้องการ มิฉะนั้นกำหนด RecordSource ไปก็ไร้ความหมายครับ (อยากให้คุณจำเกี่ยวกับเรื่องการใช้ Refresh นี้ให้ดี เพราะว่าเรามักเผลอไม่ใส่เข้าไปด้วยหลังจากเขียนคำสั่งเพื่อให้เปลี่ยนเทเบิล ความจริงหากคุณเพิ่งกำหนด RecordSource ในโพรซีเดอร์ Form_Load ก็ไม่จำเป็นต้องสั่ง Refresh หรอกนะครับ อันนี้เป็นข้อยกเว้นซ้อนลงไปอีกชั้น แต่ถ้าให้ดีจำไว้อย่างเดียวว่าต้องใช้ Refresh เสมอจะดีที่สุด เนื่องจากเรามักเปลี่ยนเทเบิลกันในโพรซีเดอร์อื่นนอกเหนือจาก Form_Load เป็นประจำ) ความจริงเพียงสามพรอพเพอร์ตี้ส์แรกนี้คุณก็สามารถติดต่อกับดาต้าเบสได้แล้ว ไม่จำเป็นต้องเซตพรอพเพอร์ตี้ส์อื่นอีก แต่ทำ ๆ ไปจะพบว่าพรอพเพอร์ตี้ส์ข้างล่างนี้มีประโยชน์กับงานของคุณอย่างยิ่งยวด



รูปที่ 3 ตัวเลือกของ RecordSource ในดาต้าเบส Biblio.MDB

RecordsetType กำหนดชนิดของ Recordset ว่าจะให้เป็น Table, Dynaset หรือว่า Snapshot โดยดีฟอลต์จะเป็น Dynaset

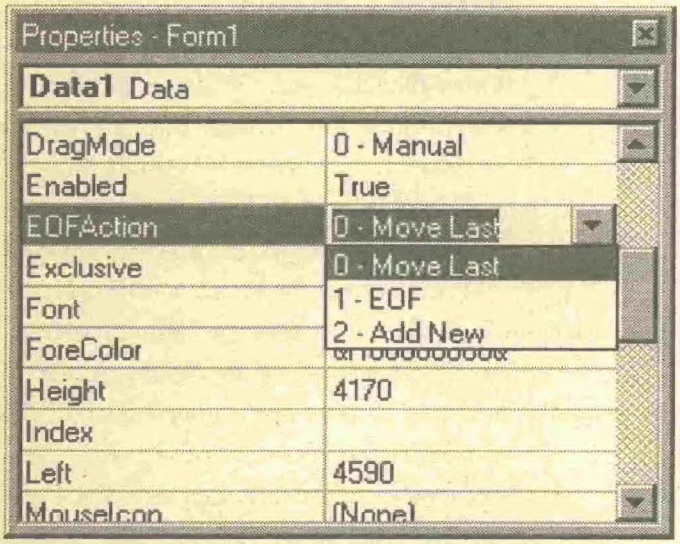
BOFAction สั่งให้ Data Control เลื่อนพอยน์เตอร์ไปยังเรคอร์ดใดหากผู้ใช้เรียกดูเรคอร์ดย้อนขึ้นมาถึงต้น Recordset มีตัวเลือกสองอย่างคือ Move First (แช่อยู่กับเรคอร์ดแรก) และ BOF (เซตพอยน์เตอร์ให้ชี้ไปที่ BOF ไม่ให้ชี้ที่เรคอร์ดใดเรคอร์ดหนึ่ง) คุณสามารถกำหนดตัวเลือกได้ในหน้าต่างพรอพเพอร์ตี้ส์โดยตรงเช่นรูปที่ 4



รูปที่ 4 ตัวเลือกของพรอพเพอร์ตี้ส์ BOFAction ในหน้าต่างพรอพเพอร์ตี้ส์

EOFAction สั่งให้ Data Control เลื่อนพอยน์เตอร์ไปยังเรคอร์ดใดหากผู้ใช้เรียกดูเรคอร์ดจนกระทั่งลงไปถึงสุด Recordset มีตัวเลือกอยู่สามตัวคือ Move Last (แช่อยู่กับเรคอร์ดสุดท้าย) EOF (เซตพอยน์เตอร์ให้ชี้ไปที่ EOF ไม่ให้ชี้

ที่เรคอร์ดใดเรคอร์ดหนึ่ง) และ Add New (เพิ่มเรคอร์ดเข้าไปใน Recordset โดยใช้วิธีการ AddNew ของ Data Control) คุณสามารถกำหนดตัวเลือกได้ในหน้าต่างพรอพเพอร์ตี้ส์โดยตรงเช่นรูปที่ 5



รูปที่ 5 ตัวเลือกของพรอพเพอร์ตี้ส์ EOFAction ในหน้าต่างพรอพเพอร์ตี้ส์

EditMode เป็นตัวบอกว่าปัจจุบันภาวะการอีดิตของเรคอร์ดเป็นเช่นไร เมื่อคุณทราบว่าภาวะการอีดิตในปัจจุบันเป็นอย่างไรก็ประยุกต์ใช้ได้กับหลายสถานการณ์ เช่น เมื่อรู้ว่าปัจจุบันไม่มีการอีดิตใด ๆ เกิดขึ้น (ค่าเป็น dbEditNone) ก็จะให้จบการทำงาน เช่น

```
Private Sub Command1_Click()
    If Data1.EditMode=dbEditNone Then
        End
    ElseIf
Data1.EditMode=dbEditInProgress Then
        MsgBox "ต้องการจัดเก็บข้อมูลปัจจุบันก่อนหรือไม่?"
    End If
End Sub
```

หมายเหตุไว้เป็นพิเศษเกี่ยวกับการใช้ค่าคงที่เช่น dbEdit None หรือ dbEditInProgress เสียหน่อย สำหรับเวอร์ชัน 4.0 นี้ คุณไม่จำเป็นต้องประกาศตัวแปรเหล่านี้ (อย่างเช่นที่เคยคัดลอกแล้วนำมาแปะจากไฟล์เช่น Constant.TXT ในเวอร์ชันก่อน) คุณใช้ได้เลยโดยตรง หากต้องการทราบว่าพรอพเพอร์ตี้ส์ใดมีค่าคงที่อะไรความหมายเป็นยังไง ก็ใช้ Object Browser ได้ เพียงกดปุ่ม F2 หรือเลือกเมนู View|Object Browser

Exclusive เมื่อมีผู้ใช้หลายคนกำลังเข้าถึงดาต้าเบสเดียวกันกับผู้ใช้แอพพลิเคชันของคุณ ปัญหาเกี่ยวกับเรื่องการร่วมใช้มันก็ตามมา ถ้าหากต้องการให้แอพพลิเคชันของคุณจับจองดาต้าเบสไว้แต่เพียงผู้เดียว คนอื่นเข้ามายุ่งเกี่ยวขณะแอพพลิเคชันกำลังทำงานไม่ได้ ก็ให้เซตค่า Exclusive นี้เป็น True (โดยดีฟอลต์จะเป็น False) แต่ว่าการเซตค่าให้เป็น True นี้จะเกิดผลก็ต่อเมื่อไม่มีใครใช้ดาต้าเบสอยู่ก่อนนะครับ หากคนอื่นเข้าถึงดาต้าเบสก่อนแอพพลิเคชันของคุณ การเซต Data1.Exclusive = True ก็จะเป็นหมันไปโดยปริยาย

ReadOnly เป็นการอนุญาตให้ผู้ใช้แอพพลิเคชันของคุณดัดแปลงแก้ไขข้อมูลในดาต้าเบสได้หรือไม่ ถ้าหากเซตเป็น True เขาก็จะทำอะไรกับข้อมูลไม่ได้เลยนอกจากเรียกดูอย่างเดียว (โดยดีฟอลต์จะเป็น False ซึ่งหมายความว่าผู้ใช้สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้)

เมธอดหลัก

Refresh เป็นการทำให้ออปเจ็กต์เช่น Recordset ถูกปรับปรุงข้อมูลให้ตรงตามจริงในดาต้าเบสปัจจุบัน ว่าตามจริงแล้ว วิธีการ Refresh ทำหน้าที่สร้างออปเจ็กต์ Recordset ขึ้นมาใหม่เลยทีเดียว ขอให้คิดถึงที่เราเปลี่ยน Record Source กันเมื่อครู่เป็นต้น หากไม่มีการสร้าง Recordset ขึ้นมาใหม่ ข้อมูลก็จะถูกปรับเปลี่ยนให้แสดงตามเทเบิลอื่นไม่ได้เลย

MoveNext เลื่อนไปยังเรคอร์ดถัดไป เช่น Data1.MoveNext

MovePrevious เลื่อนไปยังเรคอร์ดก่อนหน้า

MoveFirst กระโดดไปยังเรคอร์ดแรก

MoveLast กระโดดไปยังเรคอร์ดสุดท้าย

FindFirst ค้นหาเรคอร์ดแรก (ในกรณีที่มีหลายเรคอร์ด หรืออาจจะมีเพียงเรคอร์ดเดียวก็ได้) ที่มีค่าข้อมูลในฟิลด์ตรงกับความต้องการ เช่น Data1.FindFirst "Author='Adrian, Merv' "

FindNext ค้นหาเรคอร์ดถัดไป (ในเงื่อนไขเดียวกันกับ FindFirst)

FindPrevious ค้นหาเรคอร์ดก่อนหน้า (เงื่อนไขเดียวกันกับFindFirst)

FindLast ค้นหาเรคอร์ดสุดท้ายที่มีค่าข้อมูลในฟิลด์ตรงกับความต้องการ

อีเวนต์หลัก

Validate อีเวนต์นี้จะเกิดขึ้นขณะก่อนเรคอร์ดหนึ่ง ๆ จะกลายเป็นเรคอร์ดปัจจุบัน อันนี้มีประโยชน์ในการที่คุณจะใช้เช็คค่าตัวแปรหรือปัจจัยอะไรต่ออะไรอยู่ในภาวะสมควรให้เกิดการเปลี่ยนเรคอร์ดหรือไม่

Reposition อีเวนต์นี้เกิดขึ้นหลังจากเรคอร์ดหนึ่ง ๆ กลายเป็นเรคอร์ดปัจจุบันไปแล้ว โดยมากใช้เมื่อคุณต้องการให้เกิดฉากการณ์อย่างใดอย่างหนึ่งขึ้น โดยใช้ค่าของเรคอร์ดปัจจุบันเป็นตัวกำหนด เช่น ถ้าฟิลด์เงินเดือนของเรคอร์ดปัจจุบันมันมากกว่า 10,000 ก็ให้มีการใส่สีแดงให้ดูเด่น ถ้าต่ำกว่านี้ก็เป็นสีเขียวเป็นต้น

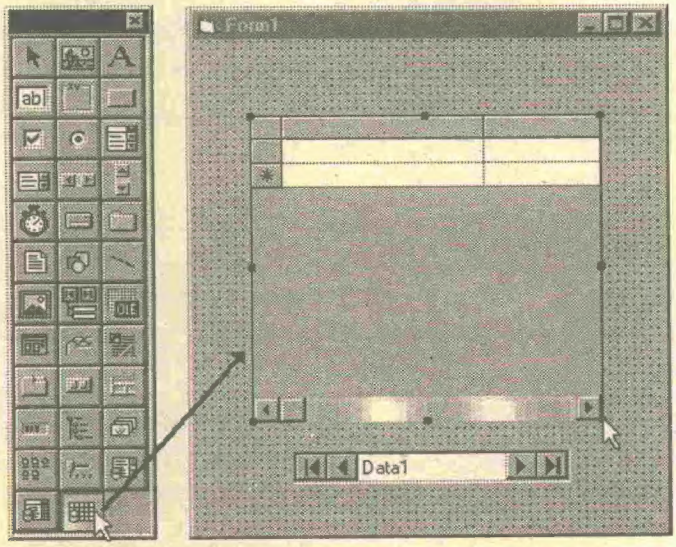
เป็นอันว่าแนะนำให้รู้จักกับ Data Control ของวิชวลเบสิก 4.0 เท่านี้ก่อน คุณจะเห็นว่าไม่มีพรอพเพอร์ตี้ส์ เมธอด หรืออีเวนต์ใดเลยที่เข้าข่ายการแสดงข้อมูลตรงจากดาต้าเบส ที่ผ่านมามันล้วนแล้วแต่เป็นเรื่องของการดึงข้อมูลหรือดำเนินการอะไรอื่นกับข้อมูลในดาต้าเบสทั้งสิ้น ไม่เกี่ยวกับการแสดงเลย ทั้งนี้ก็เพราะ Data Control ไม่มีองค์ประกอบใดรองรับการแสดงข้อมูลได้นั่นเอง ต่อไปนี้เรามาดูกันครับว่าคอนโทรลที่มีหน้าที่ในการถ่ายทอดข้อมูลจาก Data Control มาแสดงโดยตรง อันได้แก่ DBGrid, DBList และ DBCombo นั้น มีรายละเอียดและวิธีใช้อย่างไร

DBGrid (Data Bound Grid)

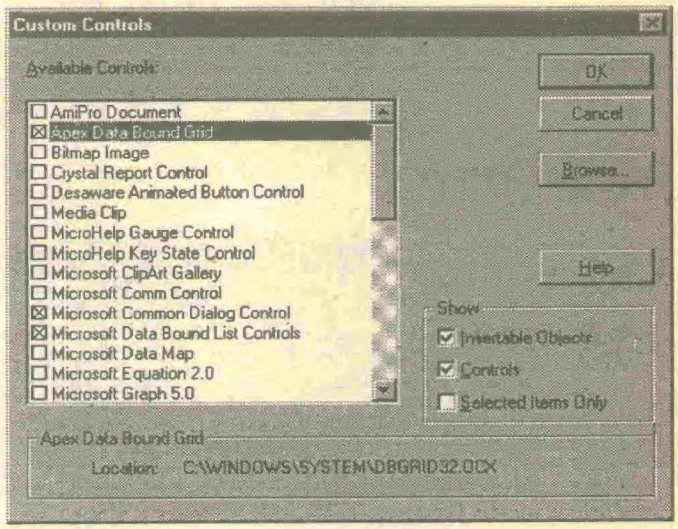
วิธีสร้างขึ้นบนฟอร์ม

เนื่องจากคอนโทรล DBGrid สมควรมีขนาดใหญ่แน่ ๆ เพื่อแสดงข้อมูลได้หลายฟิลด์และหลายเรคอร์ดพร้อมกัน จึงไม่นิยมสร้างด้วยการดับเบิลคลิ๊ก เรามักจะคลิ๊กไอคอนในทูลบ็อกซ์แล้วลากเม้าส์กำหนดขนาดไปเลยเหมือนอย่างในรูปที่ 6

โดยดีฟอลต์แล้วคุณจะเห็นไอคอนของ DBGrid ปรากฏรวมอยู่กับไอคอนทั่วไปในทูลบ็อกซ์ แต่หากต้องการย้ายออกหรือนำกลับเข้ามาก็ทำได้ด้วยการกดปุ่ม Ctrl-T หรือใช้เมนู Tools|Custom Controls จะเห็นหน้าต่าง Custom Controls ปรากฏขึ้นมาให้เลือกดังรูปที่ 7 หากคุณกาเครื่องหมายที่ช่องเช็คบ็อกซ์ไหนก็จะได้คัสตอมคอนโทรลตัวนั้นขึ้นมาในทูลบ็อกซ์หลังจากคลิ๊ก OK แล้ว และทำนองเดียวกัน หากคลิ๊กเอาเครื่องหมายออก คัสตอมคอนโทรลในทูลบ็อกซ์นั้น ๆ ก็จะหายไป ในที่นี้ Apex Data Bound Grid จะอยู่อันดับสองให้เลือก



รูปที่ 6 การสร้าง DBGrid



รูปที่ 7

หน้าที่หลัก

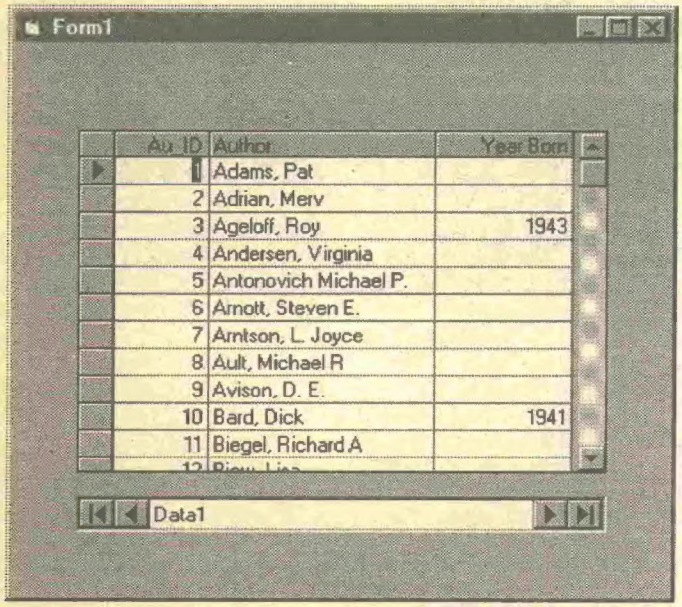
แสดงข้อมูลจากออปเจ็กต์ Recordset ได้หลายเรคอร์ดและหลายฟิลด์ในเวลาเดียวกัน ข้อมูลในที่นี้มิได้หมายเฉพาะตัวเลขหรือตัวอักษรเท่านั้น ยังรวมถึงภาพที่ถูกจัดเก็บในไบนารีฟอร์แมตอีกด้วย (ไม่ใช่เรียกขึ้นมาด้วยพรอพเพอร์ตี้ส์ Picture นะครับ ภาพต้องจัดเก็บในเทเบิลด้วยรูปฟอร์แมตไบนารีหรือเมโมเท่านั้น)

พรอพเพอร์ตี้ส์หลัก

DataSource เป็นพรอพเพอร์ตี้ส์ที่ใช้บอกว่า DBGrid จะผูกโยงอยู่กับ Data Control ตัวไหน (หนึ่งฟอร์มมีได้มากกว่าหนึ่ง Data Control ดังนั้นเราจึงต้องระบุให้แน่ชัดว่าเบาวน์คอนโทรลปัจจุบันจะเป็นของใคร) เช่นในที่นี้มีให้เลือกตัวเดียวคือ Data1 (เป็นชื่อดีฟอลต์ของ Data Control ปัจจุบันบนฟอร์ม) สำหรับพรอพเพอร์ตี้ส์นี้กำหนดได้ด้วยหน้าต่างพรอพเพอร์ตี้ส์ก่อนรันเท่านั้นครับ เปลี่ยนตอนรันด้วยโค้ดไม่ได้ หากคุณคิดจะเปลี่ยนดาต้าเบสต้องเปลี่ยนด้วยพรอพเพอร์ตี้ส์ DatabaseName ของ Data Control เลยทีเดียว

ทันทีที่ผูกโยง DBGrid เข้ากับ Data Control คุณก็พร้อมที่จะแสดงข้อมูลได้ทันทีครับ เมื่อกดปุ่ม F5 หรือคลิ๊กปุ่มรันแล้วจะเห็น DBGrid แสดงข้อมูลจากเทเบิล Authors ในดาต้าเบส Biblio.MDB ซึ่งเรากำหนดไว้ใน Data Control ทันที ดังที่เห็นในรูปที่ 8

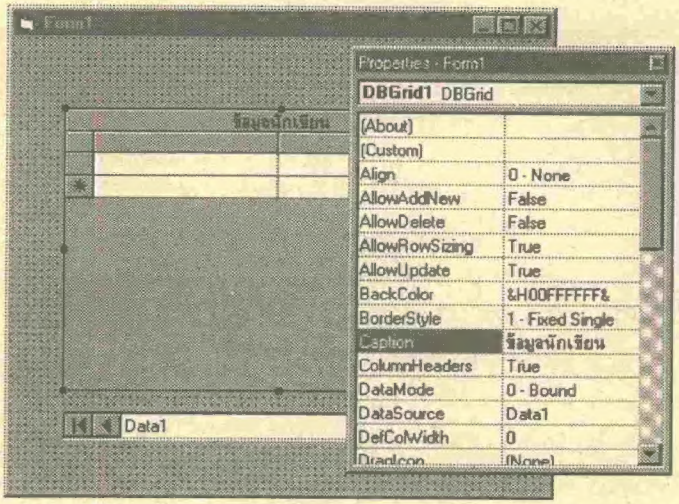
ไม่เพียงแต่ DBGrid จะทำหน้าที่แสดงข้อมูลจาก Biblio.MDB เท่านั้น บัดนี้มันยังพร้อมจะรับการแก้ไขและจัดเก็บข้อมูลใหม่จากผู้ใช้อีกด้วย วิธีนั้นก็ง่ายมาก เพียงผู้ใช้คลิ๊กโฟกัสเซลใดเซลหนึ่งก็แก้ไขได้เลย เมื่อต้องการจัดเก็บก็เพียงเลื่อนเรคอร์ดไปข้างหน้าหรือข้างหลัง (ซึ่งหากคุณจะใช้โค้ด ก็ทำได้ด้วยวิธีการ MoveNext หรือ MovePrevious ของ Data Control)



รูปที่ 8 ด้วยการกำหนดค่าพรอพเพอร์ตี้สั่ DataSource เพียงอย่างเดียว เราก็เห็นข้อมูลซึ่ง Data Control ถ่ายทอดมาให้ทันที อีกทั้งพร้อมจะรับอินพุตใหม่จากผู้ใช้อีกด้วย

Caption เป็นพรอพเพอร์ตี้สั่ที่ทำให้ตารางของคุณมีความหมายกับผู้ใช้ขึ้นมาก กล่าวคือมันจะทำให้ DBGrid สร้างไตเติลบาร์ขึ้นมาบนส่วนสูงสุดของมัน คุณไม่ต้องทำอะไรเลยนอกจากคีย์ข้อความที่ต้องการในช่องรับ Value ของพรอพเพอร์ตี้สั่ Caption นี้ เช่นตัวอย่างรูปที่ 9

Col ใช้ในการเซตหรือคืนค่าคอลัมน์ปัจจุบัน (เหมือนกับ Grid.Col) เช่น DBGrid1.Col = 10 หรือ Index=DBGrid1.Col



รูปที่ 9 เขียนในช่องรับ Value ของ Caption อย่างไร คุณก็จะได้ไตเติลบาร์ของ DBGrid ขึ้นมาอย่างนั้น

Row ใช้ในการเซตหรือคืนค่าบรรทัดปัจจุบัน (เหมือนกับ Grid.Row) เช่น DBGrid1.Row = 5 หรือ Index=DBGrid1.Row

Bookmark Bookmark เป็นสตริง (ไม่ใช่เลขจำนวนเต็ม) ที่ระบบสร้างขึ้นเพื่อชี้ว่าตอนนี้มีเรคอร์ดใดเป็นปัจจุบัน สำหรับพรอพเพอร์ตี้สั่ Bookmark ของ DBGrid นั้นจะใช้อ้างค่าเรคอร์ดปัจจุบันมาเก็บไว้ใช้ เช่น varCurRecord= DBGrid1.Bookmark (ตัวแปร varCurRecord ต้องประกาศเป็น String หรือ Variant นะครับ ไม่ใช่ Integer) เมื่อเกิดการเปลี่ยนเรคอร์ดไปเป็นอื่นแล้วต้องการนำเรคอร์ดเดิมกลับมาใหม่ก็กำหนดว่า DBGrid1.Bookmark=varCurRecord

ColIndex ใช้ในการเซตตำแหน่งหรือเก็บค่าคอลัมน์ปัจจุบัน ทำนองเดียวกันกับที่พรอพเพอร์ตี้สั่ Bookmark มีไว้เซตหรืออ้างตำแหน่งเรคอร์ดปัจจุบัน

เมธอดหลัก

Refresh ปรับปรุงข้อมูลให้ตรงตามดาต้าเบสปัจจุบัน

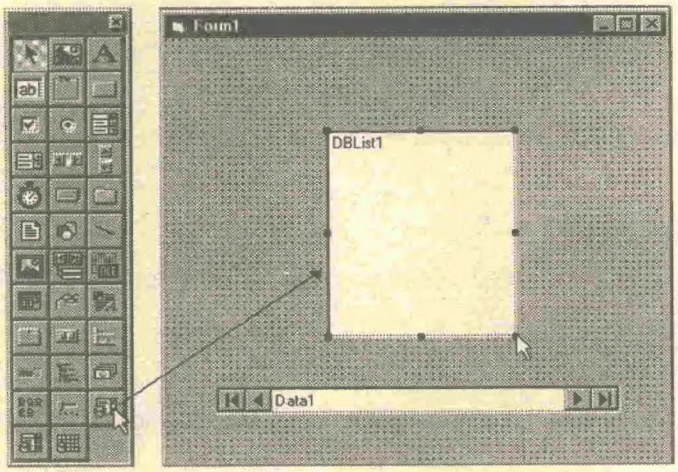
อีเวนต์หลัก

RowColChange เป็นอีเวนต์ที่เกิดขึ้นหลังจากมีการเปลี่ยนโฟกัสตำแหน่งเซลปัจจุบันไปเป็นเซลใหม่ โดยในอีเวนต์นี้จะมีอาร์กิวเมนต์เป็น LastRow และ LastCol ไว้ให้ เผื่อคุณอยากนำโฟกัสเซลเก่ากลับมาจะได้ง่ายเข้า ไม่ต้องประกาศตัวแปรไว้เก็บค่าเดิมเอาเอง

DBList

วิธีสร้างขึ้นบนฟอร์ม

คลิ๊กไอคอนและลากเม้าส์บนฟอร์มดังรูปที่ 10



รูปที่ 10 การสร้าง DBList

หากต้องการนำคัสตอมคอนโทรลตัวนี้ออกจากทูลบ็อกซ์หรือนำเข้ามารวมในทูลบ็อกซ์ใหม่ ให้ใช้หน้าต่าง Custom Controls และกาเครื่องหมายที่กล่องเช็คบ็อกซ์ของ Micorsoft Data Bound List Controls สังเกตคำว่า Controls ในที่นี้มี s นะครับ เขาไม่ได้เขียนผิดหรอก เพราะการนำมาซึ่งคัสตอมคอนโทรล "ชนิด" Data Bound List นี้ คุณจะได้ทั้ง DBList และ DBCombo

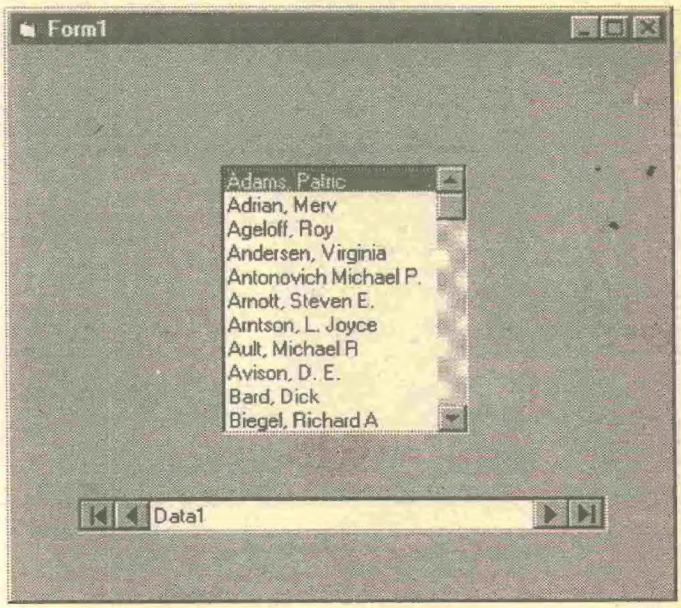
หน้าที่หลัก

แสดงข้อมูลหนึ่งฟิลด์จากออปเจ็กต์ Recordset ได้หลายเรคอร์ดในเวลาเดียวกัน หากคุณใช้ Listbox ธรรมดา แม้จะมีความสามารถเป็นเบาวน์คอนโทรลได้ กว่าจะแสดงข้อมูลจาก Data Control มันก็ต้องอาศัยวิธีการ AddItem ซึ่งคุณต้องเขียนเอาเองอยู่ดี ส่วน DBList นี้ เพียงเมื่อคุณเซตพรอพเพอร์ตี้ส์ในโหมดก่อนรันให้ถูกต้อง ข้อมูลก็พร้อมจะแสดงทันทีโดยคุณไม่ต้องเสียเวลาโค้ดอะไรเลย

พรอพเพอร์ตี้ส์หลัก

RowSource สองพรอพเพอร์ตี้ส์ที่สำคัญที่สุดไม่ใช่ DataSource และ DataField เหมือนอย่างเบาน์คอนโทรลตัวอื่นเสียแล้วครับ แม้คุณจะพบว่ามีทั้ง DataSource และ DataField รวมอยู่ในกลุ่มพรอพเพอร์ตี้ส์ของ DBList ก็ตาม หากคุณเซตพรอพเพอร์ตี้ส์สองตัวนั้นแล้วรันโดยหวังจะได้เห็นการแสดงข้อมูลทาง DBList ทันทีก็คงต้องผิดหวัง เพื่อที่จะแสดงข้อมูลได้ในทันทีทันใด คุณต้องใช้พรอพเพอร์ตี้ส์ RowSource เป็นตัวระบุว่าจะแสดงข้อมูลในออปเจ็กต์ Recordset ของ Data Control ตัวใด เช่นในที่นี้ก็ต้องเลือก Data1 เหมือนเคยเพราะเรายังไม่ได้เพิ่ม Data Control เข้าไปใหม่

ListField พรอพเพอร์ตี้ส์นี้เป็นตัวระบุว่าจะเลือกข้อมูลฟิลด์ใดมาแสดง เพียงเมื่อกำหนดค่าให้พรอพเพอร์ตี้ส์นี้เสร็จ คุณก็พร้อมแล้วที่จะแสดงข้อมูลได้ทันทีดังรูปที่ 11 (เวลาจำควรจำเป็นคู่นะครับ คือให้จำ RowSource คู่กับ ListField เหมือนอย่างที่คุณเคยจำคู่ของ DataSource และ DataField)



รูปที่ 11 แสดงข้อมูลจาก Data Control ด้วย DBList

BoundText Text ของบรรทัดในลิสต์ตำแหน่งปัจจุบัน

เมธอดหลัก

Refresh ปรับปรุงข้อมูลให้ตรงตามจริงในปัจจุบัน

อีเวนต์หลัก

Click ถูกเรียกเมื่อผู้ใช้คลิ๊กเม้าส์ลงบนบรรทัดใดบรรทัดหนึ่ง เรามักต้องใช้อีเวนต์นี้ เนื่องจากผู้ใช้มักเกี่ยวข้องกับคอนโทรลจำพวกลิสต์โดยการคลิ๊กเลือกข้อมูลบรรทัดใดบรรทัดหนึ่งเพื่อนำข้อมูลไปใช้อ้างหรือเพื่อเปลี่ยนแปลง เช่น

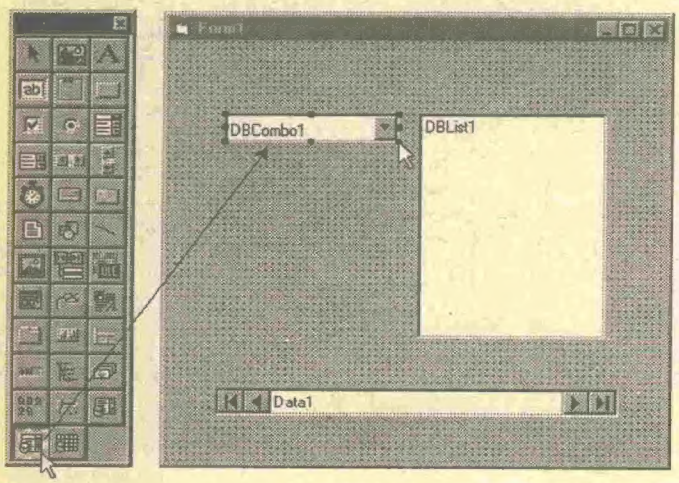
```
Private Sub DBList1_Click ()
      Label1.Caption = DBList1.BoundText
End Sub
```

ซึ่งหมายความว่าทันทีที่ผู้ใช้คลิ๊กเลือกบรรทัดข้อมูลใน DBList จะเห็นข้อมูลในบรรทัดดังกล่าวไปปรากฏอยู่ใน Label ทันที

DBCombo

วิธีสร้างขึ้นบนฟอร์ม

คลิ๊กไอคอนและลากเม้าส์บนฟอร์มดังรูปที่ 12



รูปที่ 12 การสร้าง DBCombo

การสร้าง DBCombo

อย่างที่บอกไว้แล้วในหัวข้อ DBList ว่าเป็นคัสตอมคอนโทรลที่มาและไปพร้อมกับ DBList เนื่องจากสองตัวนี้อยู่ในการเลือกเช็คบ็อกซ์ชื่อ Microsoft Data Bound List Controls เหมือนกัน หากคุณมี DBList ได้ก็ต้องมี DBCombo ด้วย

หน้าที่หลัก

เหมือนกับ DBList ต่างกันตรงที่จะมีช่องรับข้อมูลเพิ่มขึ้นมา

พรอพเพอร์ตี้ส์หลัก

RowSource กำหนดชื่อ Data Control ซึ่งเราต้องการให้ DBCombo แสดงข้อมูลออกมา

ListField ระบุชื่อฟิลด์ที่ต้องการให้แสดงข้อมูล เพียงเลือกสองพรอพเพอร์ตี้ส์นี้คุณสามารถแสดงข้อมูลได้ทันที แต่ยังไม่อนุญาตให้ผู้ใช้ป้อนอินพุตด้วยช่องรับข้อมูลประจำคอนโทรล

DataSource กำหนดชื่อ Data Control ซึ่งเป็นที่หมายซึ่งเราสามารถดำเนินการกับข้อมูลได้

DataField กำหนดชื่อฟิลด์ที่เราต้องการดำเนินการ โดยเฉพาะในเรื่องของการเพิ่มและแก้ไขข้อมูล

BoundText Text ของบรรทัดในลิสต์ตำแหน่งปัจจุบัน

เมธอดหลัก

Refresh ปรับปรุงข้อมูลให้ตรงตามจริงในปัจจุบัน

อีเวนต์หลัก

Click ถูกเรียกเมื่อเกิดการคลิ๊กบรรทัดข้อมูลจากผู้ใช้

สำหรับ DBList และ DBComo ของไมโครซอฟต์นั้นออกจะใช้ยากอยู่สักนิด เอาไว้โอกาสต่อไปผมจะนำเรื่องของการประยุกต์ใช้มาเสนอ ตอนนี้รู้จักเรื่องหลักพอสังเขปก่อน สวัสดีครับ